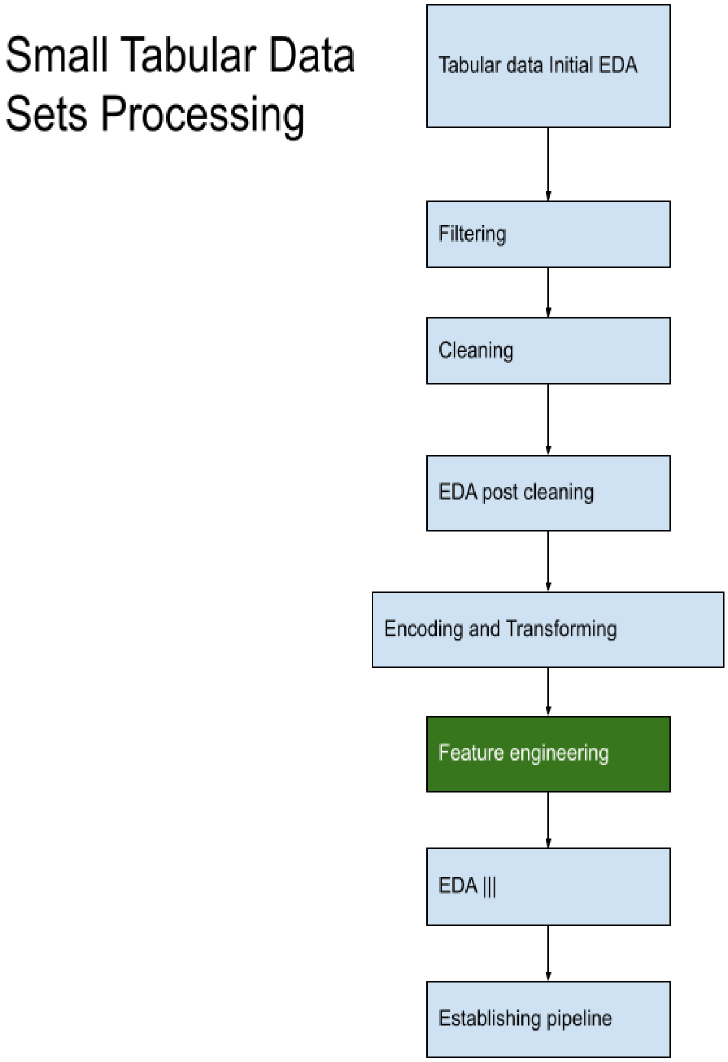
Feature Engineering Excluding univariate operators (Encoding, Transformations)


Small Tabular Data Sets Processing

1. Interaction Terms: Create new features by combining existing ones. For instance, division, multiplication , min/max (if they are on the same scale , else use the normalized or other transform)
    1.1. Basic Multiplicative Interactions:
        1.1.1. Creation: Multiply two or more features together.
        1.1.2. Example: If you have features 'Age' and 'Annual Income' in a financial dataset, you can create an interaction term by multiplying them (`Age * Annual Income`). This new feature could represent the interaction between a person's age and income level.
    1.2. Categorical and Continuous Interactions:

1.2.1. Creation: First, encode the categorical variable (e.g., using one-hot encoding), then multiply each encoded category by a continuous variable.

1.2.2. Example: In a housing dataset, you might have 'HasGarage' (a binary variable) and 'HouseSize' (a continuous variable). You can create an interaction term by multiplying these (`HasGarage * HouseSize`), capturing the potential combined effect of having a garage and the size of the house on the house's value.

1.3. Polynomial Interactions:

1.3.1. Creation: Create higher-order terms (squared, cubic, etc.) for continuous variables.

1.3.2. Example: With a feature 'YearsOfExperience', you could create squared (`YearsOfExperience^2`) or cubic terms (`YearsOfExperience^3`) to capture non-linear relationships.

1.4. Combinations of More Than Two Features:

1.4.1. Creation: Extend the concept of multiplicative interactions to more than two features.

1.4.2. Example: In a retail dataset, an interaction term could be created from 'Number of Store Visits', 'Average Purchase Value', and 'Number of Online Logins' by multiplying these three features together.

1.5. Categorical-Categorical Interactions:

1.5.1. Creation: After encoding categorical variables, create interaction terms by multiplying these encoded features.

1.5.2. Example: In a dataset with 'Vehicle Type' and 'Fuel Type', create an interaction term by multiplying the encoded variables. This could reveal specific combinations of vehicle and fuel types that correlate strongly with outcomes like resale value.

2. Temporal Features: If there are date or time features, you can extract information like day of the week, month, or duration from a particular reference point (e.g., duration since first becoming a customer), create relevant time series operators such as moving average etc.

2.1.1. Extraction of Temporal Features

2.1.2. Breaking Down Date-Time:

2.1.2.1. Extract components like year, month, day, day of the week, hour, minute, and second.

2.1.2.2. Example: From a timestamp in a sales dataset, extract 'Year', 'Month', 'Day', 'Hour' to analyze sales patterns over time.

2.1.3. Special Dates and Events:

2.1.3.1. Extract information about holidays, weekends, or special events that might affect the data.

2.1.3.2. Example: In a retail dataset, flag days as 'Holiday' or 'Black Friday' to capture their impact on sales.

2.1.4. Time Since a Particular Event:

2.1.4.1. Calculate the time elapsed since a significant event.

2.1.4.2. Example: In customer churn analysis, calculate the number of days since the last purchase.

2.1.5. Seasonality Features:

2.1.5.1. Create features to capture seasonal effects like quarters or seasons.

2.1.5.2. Example: Add a feature 'Season' indicating 'Spring', 'Summer', 'Fall', 'Winter' in a dataset related to agricultural production.

2.1.6. Utilization of Temporal Features

2.1.7. Trend Analysis:

2.1.7.1. Use extracted features to analyze trends over time.

2.1.7.2. Example: Analyze sales trends over months and years to understand growth patterns.

2.1.8. Cyclical Relationships:

2.1.8.1. Certain temporal components are cyclical, like hours in a day or days in a week.

2.1.8.2. Example: Predicting energy consumption might require capturing the cyclical nature of hours in a day.

2.1.9. Cyclical Encoding for Time-based Features

2.1.10. Cyclical features, such as hours of the day or days of the week, don't have a natural start or end point, and their relationships can be better captured through cyclical encoding:

2.1.11. Sine and Cosine Transformations:

2.1.11.1. Transform cyclical features using sine and cosine functions to preserve their cyclical nature.

2.1.11.2. Example: Encode 'HourOfDay' using sine and cosine to ensure that the model understands that hour 23 is close to hour 0.

2.1.12. Example of Cyclical Encoding:

2.1.12.1. Suppose you have an 'HourOfDay' feature ranging from 0 to 23. You can encode it using:

2.1.12.1.1. `sin(2 * π * HourOfDay / 24)` and `cos(2 * π * HourOfDay / 24)`

2.1.12.2. This transformation maps each hour to a point on a circle, accurately representing the cyclical nature.

2.1.13. Considerations in Utilizing Temporal Features

2.1.14. Overfitting to Specific Dates/Times: Be cautious about overfitting to specific dates or times unless they are consistently relevant (like annual holidays).

2.1.15. Time Zone Adjustments: Ensure that time data is consistent, especially if it's collected across different time zones.

2.1.16. Handling Missing or Irregular Timestamps: Decide how to handle missing values or irregular time intervals in your time-based data.

2.1.17. In summary, effectively extracting and utilizing temporal features means understanding the time-related patterns and cycles in your data and using various techniques to transform these time elements into meaningful features for your models. This can significantly enhance the performance of time-sensitive predictive models.

3. Segmentation Features, using clustering algorithms to create features:

3.1. Choose Relevant Features

3.2. Normalize the Data

3.3. Since clustering algorithms like K-Means are sensitive to the scale of the data, it's important to normalize the features.

3.4. Apply a Clustering Algorithm

3.5. The number of clusters can be determined using methods like the Elbow Method or Silhouette Analysis.

3.6. Assign the cluster labels to each instance in your dataset. These labels represent the segment of each instance.

4. Grouping and Aggregating , Grouping by : categorical variable, binned numerical variable, properties of textual features :

4.1. Group Variability Feature" involves quantifying the variability or dispersion of a certain attribute within groups defined by another attribute. This can be done using statistical measures like the standard deviation, range, or variance or entropy within each group.

4.2. Interaction with group variability feature
4.3. Operations relative to the group properties like quantile, ranking etc